AMENDMENTS TO THE CLAIMS

Please accept amended Claims 1-3 and 5-9 as follows:

1. (Currently Amended) A computer-implemented method for processing a first instruction ~~form~~ set and a second instruction ~~form~~ set ~~of an instruction set~~ in a processor comprising the steps of:

providing a program of instructions comprising a plurality of instructions of the first instruction ~~form~~ set and a plurality of instructions of the second instruction ~~form~~ set, wherein the plurality of instructions of the first instruction ~~form~~ set are decoded by a decoder in an execution pipeline and the plurality of instructions of the second instruction ~~form~~ set are predecoded by a compiler;

storing the plurality of instructions of the second ~~form~~ set in a plurality of buffers proximate to a plurality of execution units;

executing at least one instruction of the first instruction ~~form~~ set in response to a first counter; and

executing at least one instruction of the second instruction ~~form~~ set in response to at least a second counter, wherein the second counter is invoked by a branch instruction of the first instruction ~~form~~ set,

wherein the step of executing at least one instruction of the second instruction ~~form~~ set further comprises the steps of

de-gating a plurality of execution queues storing the plurality of instructions of the first instruction ~~form~~ set, and

pausing a fetching of the first instruction ~~form~~ set from a memory.

2. (Currently Amended) The method of claim 1, wherein the instructions of the first ~~form~~ set and instructions of the second ~~form~~ set are generated by a compiler, wherein instructions of the second ~~form~~ set are statically loaded into the plurality of buffers as control signals ready for execution.

3. (Currently Amended) The method of claim 2, wherein instructions of the second ~~form~~ set are more frequently executed than instructions of the first ~~form~~ set.

4. (Cancelled)

5. (Currently Amended) The method of claim 1, wherein the step of executing at least one instruction of the second instruction ~~form~~ set further comprises the steps of:

fetching at least one instruction of the second instruction ~~form~~ set from a buffer of the plurality of buffers; and

sequencing the at least one instruction of the second instruction ~~form~~ set to the execution units.

6. (Currently Amended) The method of claim 1, wherein the second instruction ~~form~~ set is a logical subset of the first instruction ~~form~~ set.

7. (Currently Amended) The method of claim 1, wherein the step of executing at least one instruction of the first instruction ~~form~~ set further comprises the steps of:

fetching an instruction of the first form ~~form~~ set a memory;

decoding the instruction; and

issuing the decoded instruction to at least one execution unit.

8. (Currently Amended) The method of claim 1, wherein a return to fetching of the first instruction ~~form~~ set is signaled by a switch bit in a buffer of a branch unit storing instructions of the second ~~form~~ set.

9. (Currently Amended) The method of claim 1, wherein a return to fetching of the first instruction ~~form~~ set is signaled by a return instruction of the second instruction ~~form~~ set stored in a buffer of a branch unit.

10. (Previously Presented) The method of claim 1, wherein each execution unit is associated with a different buffer of the plurality of buffers.

11. (Previously Presented) A processor for processing a program of instructions comprising instructions of a first instruction form and a second instruction form comprising:

a plurality of execution units for receiving instructions;

a branch unit connected to an instruction fetch unit for the first instruction form and a sequencer for the second instruction form, wherein the sequencer controls a plurality of gates connected between a plurality of execution queues for storing decoded instructions of the first instruction form and the plurality of execution units;

a decode unit for decoding instructions of the first instruction form into control signals for the execution units; and a plurality of buffers, proximate to the execution units, for storing predecoded instructions of the second instruction form.

12. (Previously Presented) The processor of claim 11, wherein the instructions of the first form and instructions of the second form are generated based on execution frequency, wherein instructions of the second form are executed more frequently than instructions of the first form.

13. (Previously Presented) The processor of claim 11, wherein the sequencer, engaged by the branch unit, addresses the predecoded instructions of the second instruction form stored in the buffers and sequences predecoded instructions of the second instruction form to the execution unit.

14. (Cancelled)

15. (Previously Presented) The processor of claim 11, wherein each execution unit is connected to a corresponding buffer of the plurality of buffers.

16. (Original) The processor of claim 11, wherein the branch unit switches the processor from the first instruction form to the second instruction form in response to a branch instruction of the first instruction form.

17. (Original) The processor of claim 11, wherein the branch unit switches the processor from the second instruction form to the first instruction form in response to a branch instruction of the second instruction form.

18. (Previously Presented) The processor of claim 11, wherein a switch bit in a buffer of the plurality of buffers connected to the branch unit signals the sequencer to stop fetching from the buffers and enables instruction fetching from a memory storing instructions of the first instruction form.

19. (Previously Presented) The processor of claim 11, wherein an execution bandwidth of the execution units is larger than a fetch/issue bandwidth of the first form.

20. (Previously Presented) The processor of claim 11, wherein the second instruction form is a logical subset of the first instruction form, wherein the predecoded instructions of the second instruction form are statically stored in the plurality of buffers, and wherein the predecoded instructions of the second instruction form are control signals generated by a compiler and are not decoded during a runtime of the program.

21. (Previously Presented) A processor for processing a first instruction form and a second instruction form of an instruction set comprising:

a plurality of execution units for receiving instructions;

a branch unit connected to an instruction fetch unit for the first instruction form and a sequencer for the second instruction form, wherein the branch unit switches the processor from the first instruction form to the second instruction form in response to a branch instruction of the first instruction form and switches the processor from the second instruction form to the first instruction form in response to a branch instruction of the second instruction form;

a decode unit adapted to decode instructions of the first instruction form into control signals for the execution units;

an issue unit adapted to sequence decoded instructions of the first instruction form;

a plurality of buffers, proximate to the execution units, for statically storing predecoded instructions of the second instruction form, wherein each execution unit is connected to a corresponding buffer of the plurality of buffers; and

the sequencer, engaged by the branch unit, adapted to fetch the predecoded instructions and sequence the predecoded instructions of the second instruction form, wherein the sequencer is connected to a plurality of gates connected between a plurality of execution queues adapted to store the decoded instructions of the first instruction form and the plurality of execution units, the sequencer further adapted to control the gates.